

# Conditional Diffusion Models for IR

Yuanzhi Zhu  
SP at CVL

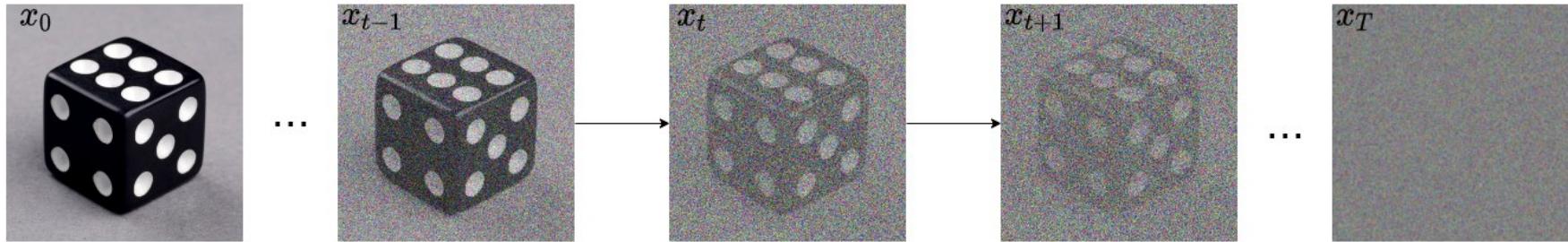
# Content

---

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

# DDPM: Denoising Diffusion Probabilistic Models

$$x_0 \sim q(x_0) \quad \longrightarrow \quad q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) \quad \longrightarrow \quad q(x_{1:T} | x_0) = \prod_{t=1}^T q(x_t | x_{t-1})$$



$$p_\theta(x_0) = \int p(x_T) \prod_{i=1}^T p_\theta(x_{t-1} | x_t) dx_{1:T} \quad \longleftarrow \quad p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad \longleftarrow \quad x_T \sim \mathcal{N}(0, I)$$

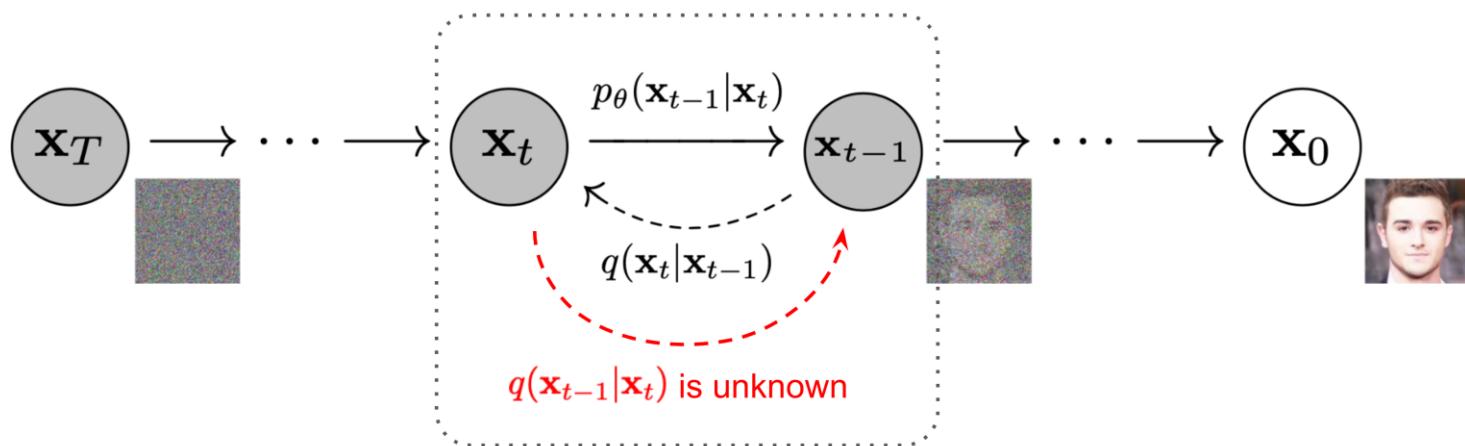
# DDPM: Denoising Diffusion Probabilistic Models

True data dist. :  $x_0 \sim q(x_0)$

Forward process:  $q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1})$  Markov Assumption

Reverse process:  $p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t)$

Use variational lower bound



# DDPM: Denoising Diffusion Probabilistic Models

## Forward Diffusion Process

$$q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1})$$

Each Step

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \underbrace{\sqrt{1 - \beta_t}x_{t-1}}_{\text{norm invariant}}, \beta_t \mathbf{I}) \quad \text{or} \quad x_t = \sqrt{1 - \beta_t}x_{t-1} + \sqrt{\beta_t}z_{t-1}$$

variance schedule  $\beta_t$  controls the diffusion processing

For arbitrary  $t$

$$\begin{aligned} q(x_t|x_0) &= \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}) & \alpha_t = 1 - \beta_t \text{ and } \bar{\alpha}_t = \prod_{i=1}^T \alpha_i \\ x_t &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \cdot z_t \end{aligned}$$

# DDPM: Denoising Diffusion Probabilistic Models

## Reverse Diffusion Process

if  $\beta_t$  is small enough,  $q(x_{t-1}|x_t)$  will also be Gaussian

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t))$$

Reverse when condition on  $x_0$

$$\begin{aligned} q(x_{t-1}|x_t, x_0) &= q(x_t|x_{t-1}, x_0) \frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} \\ &\propto \exp \left( -\frac{1}{2} \left( \frac{(x_t - \sqrt{\alpha_t} x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\bar{\alpha}_{t-1}} x_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\bar{\alpha}_t} x_0)^2}{1-\bar{\alpha}_t} \right) \right) \\ &= \exp \left( -\frac{1}{2} \left( \left( \frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}} \right) x_{t-1}^2 - \left( \frac{2\sqrt{\alpha_t}}{\beta_t} x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}} x_0 \right) x_{t-1} + C(x_t, x_0) \right) \right) \\ &\rightarrow q(x_{t-1}|x_t, x_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\boldsymbol{\beta}}_t \mathbf{I}) \end{aligned}$$

$$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0 = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \mathbf{z}_t \right)$$

$\mathbf{z}_t$  is the noise between  $x_t$  and  $x_0$

$$\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$$

# DDPM: Denoising Diffusion Probabilistic Models

## Negative Log Likelihood to Variational Lower Bound

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{\text{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\ &= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)} \right] \\ &= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_q \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \log p_\theta(\mathbf{x}_0) \right] \\ &= \mathbb{E}_q \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \\ \text{Let } L_{\text{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \geq -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0) \end{aligned}$$

# DDPM: Denoising Diffusion Probabilistic Models

## Parameterization for Training Loss

$$\begin{aligned}
 L_{\text{VLLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[ \log \frac{q(\mathbf{x}_{1:T} | \mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] = \mathbb{E}_q \left[ \log \frac{\prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)} \right] \\
 &= \mathbb{E}_q \left[ \log \frac{q(\mathbf{x}_T | \mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \right] \xrightarrow{\text{Known}} \\
 &= \mathbb{E}_q \underbrace{[D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p_\theta(\mathbf{x}_T))]}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_t} \underbrace{- \log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}_{L_0}
 \end{aligned}$$

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t)) \quad \begin{cases} \boldsymbol{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_\theta(x_t, t)) \\ \boldsymbol{\Sigma}_\theta(x_t, t) = \sigma_t^2 \mathbf{I} \end{cases}$$

## Model The Noise(Residual)

$$L_t^{\text{simple}} = \mathbb{E}_{x_0, z_t} \left[ \|z_t - z_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z_t, t)\|^2 \right] \quad x_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot z_t$$

# DDPM: Denoising Diffusion Probabilistic Models

Nonetheless, it is just **another parameterization** of  $p_\theta(x_{t-1}|x_t)$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t))$$

$$\begin{cases} \boldsymbol{\mu}_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}z_\theta(x_t, t)) \\ \boldsymbol{\Sigma}_\theta(x_t, t) = \sigma_t^2 \mathbf{I} \end{cases}$$

two options †  $\sigma^2 = \begin{cases} \beta_t \\ \frac{1 - \hat{\alpha}_{t-1}}{1 - \hat{\alpha}_t} \beta_t \end{cases}$

---

## Algorithm 1 Training

```
1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
       $\nabla_\theta \|\epsilon - \mathbf{z}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2$ 
6: until converged
```

---

## Algorithm 2 Sampling

---

```
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{z}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 
```

---

† Covariance has analytical optimal form ([Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models](#))

# DDIM: Denoising Diffusion Implicit Models\*

## Variational Inference for *Non-Markovian* Forward Processes

$$-\log p_\theta(\mathbf{x}_0) \leq -\log p_\theta(\mathbf{x}_0) + D_{\text{KL}}(q(\mathbf{x}_{1:T}|\mathbf{x}_0) \| p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0))$$

Model this directly

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0) \prod_{t=2}^T q_\sigma(x_{t-1}|x_t, x_0)$$

**Reverse Process:** deterministic given  $x_t, x_0$

$$\begin{aligned} x_{t-1} &= \sqrt{\alpha_{t-1}} x_0 + \sqrt{1 - \alpha_{t-1}} \cdot \epsilon_{t-1} = \sqrt{\alpha_{t-1}} x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_t + \sigma_t \epsilon \\ q_\sigma(x_{t-1}|x_t, x_0) &= \mathcal{N}\left(\sqrt{\alpha_{t-1}} x_0 + \sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\alpha_t} x_0}{\sqrt{1 - \alpha_t}}, \sigma_t^2 \mathbf{I}\right) \end{aligned}$$

**Forward:** still Gaussian (non-Markovian)

$$q_\sigma(x_t|x_{t-1}, x_0) = \frac{q_\sigma(x_{t-1}|x_t, x_0)q_\sigma(x_t|x_0)}{q_\sigma(x_{t-1}|x_0)}$$

[2010.02502] Denoising Diffusion Implicit Models (arxiv.org)

$\alpha_t$  in DDIM is  $\bar{\alpha}_t$  in DDPM.

# DDIM: Denoising Diffusion Implicit Models

Given: noisy observation  $x_t$

Prediction of the corresponding  $\hat{x}_0(x_t) = f_\theta^{(t)}(x_t) = (x_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(x_t)) / \sqrt{\alpha_t}$

Model difference between  $x_0$  and  $x_t$

$$p_\theta^{(t)}(x_{t-1}|x_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(x_1), \sigma_1^2 I) & \text{if } t = 1 \\ q_\sigma(x_{t-1}|x_t, f_\theta^{(t)}(x_t)) & \text{otherwise} \end{cases}$$

**Variational Inference Objective** (equivalent to objective in DDPM for certain weights)

$$\begin{aligned} J_\sigma(\epsilon_\theta) &:= \mathbb{E}_{\mathbf{x}_{0:T} \sim q_\sigma(\mathbf{x}_{0:T})} [\log q_\sigma(\mathbf{x}_{1:T}|\mathbf{x}_0) - \log p_\theta(\mathbf{x}_{0:T})] \\ &= \mathbb{E}_{\mathbf{x}_{0:T} \sim q_\sigma(\mathbf{x}_{0:T})} \left[ \log q_\sigma(\mathbf{x}_T|\mathbf{x}_0) + \sum_{t=2}^T \log q_\sigma(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) - \sum_{t=1}^T \log p_\theta^{(t)}(\mathbf{x}_{t-1}|\mathbf{x}_t) - \log p_\theta(\mathbf{x}_T) \right] \end{aligned}$$

**Surrogate Objective**  $L_t^{\text{simple}} = \mathbb{E}_{x_0, z_t} \left[ \|z_t - z_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} z_t, t)\|^2 \right]$  Same as in DDPM!

# DDIM: Denoising Diffusion Implicit Models

**Sampling** from Generalized Generative Processes  $p_\theta^{(t)}(x_{t-1}|x_t)$

$$x_{t-1} = \underbrace{\sqrt{\alpha_{t-1}} \left( \frac{x_t - \sqrt{1 - \alpha_t} \epsilon_\theta^{(t)}(x_t)}{\sqrt{\alpha_t}} \right)}_{\text{predicted } x_0} + \underbrace{\sqrt{1 - \alpha_{t-1} - \sigma_t^2} \cdot \epsilon_\theta^{(t)}(x_t)}_{\text{direction pointing to } x_t} + \underbrace{\sigma_t \epsilon_t}_{\text{random noise}}$$

$$\sigma_t := \eta \sqrt{(1 - \alpha_{t-1}) / (1 - \alpha_t)} \sqrt{1 - \alpha_t / \alpha_{t-1}}$$

- **DDPM:**  $\eta = 1$  (forward process becomes Markovian (different noise schedule from vanilla DDPM))
- **DDIM:**  $\eta = 0$  (forward process becomes deterministic)

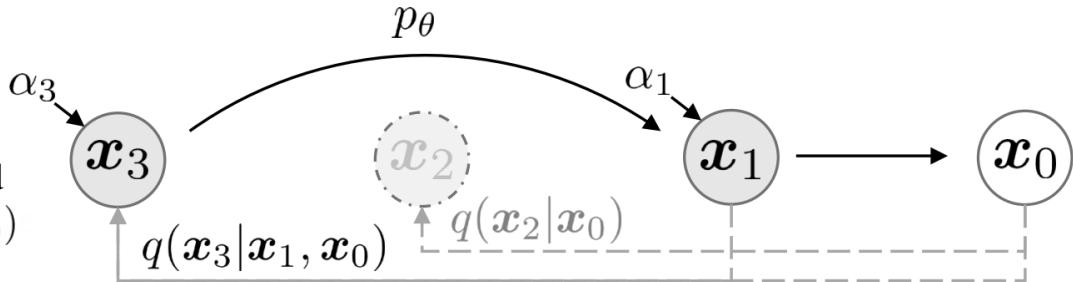
Table 1: CIFAR10 and CelebA image generation measured in FID.  $\eta = 1.0$  and  $\hat{\sigma}$  are cases of **DDPM** (although Ho et al. (2020) only considered  $T = 1000$  steps, and  $S < T$  can be seen as simulating DDPMs trained with  $S$  steps), and  $\eta = 0.0$  indicates **DDIM**.

$S$	CIFAR10 ( $32 \times 32$ )					CelebA ( $64 \times 64$ )				
	10	20	50	100	1000	10	20	50	100	1000
0.0	<b>13.36</b>	<b>6.84</b>	<b>4.67</b>	<b>4.16</b>	4.04	<b>17.33</b>	<b>13.73</b>	<b>9.17</b>	<b>6.53</b>	3.51
0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
$\hat{\sigma}$	367.43	133.37	32.72	9.99	<b>3.17</b>	299.71	183.83	71.71	45.20	<b>3.26</b>

# DDIM: Denoising Diffusion Implicit Models

## Accelerated Generation Processes

Denoising surrogate objective does not depend on the specific forward procedure  $q_\sigma(x_{t-1}|x_0)$



Consider the forward process as defined on a subset  $\tau = [\tau_1, \tau_2, \dots, \tau_{\dim(\tau)}] \subset [1, 2, \dots, T]$

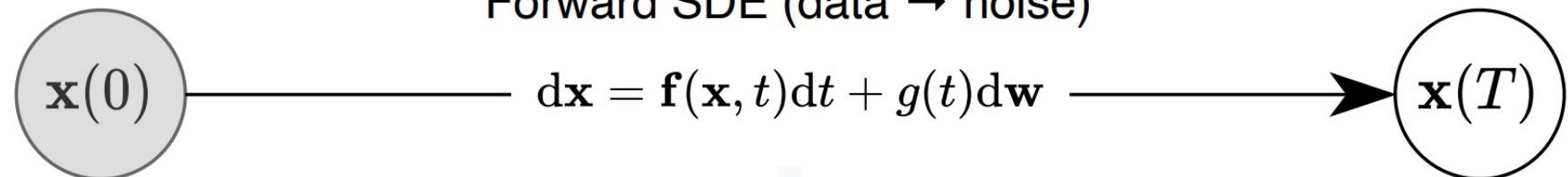
$$q_{\sigma, \tau}(x_{\tau_{i-1}}|x_{\tau_t}, x_0) = \mathcal{N}\left(x_{\tau_{i-1}}; \sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \frac{x_{\tau_i} - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2 \mathbf{I}\right)$$

The generative process now samples latent variables according to  $\text{reversed}(\tau)$ , which we term (sampling) *trajectory*

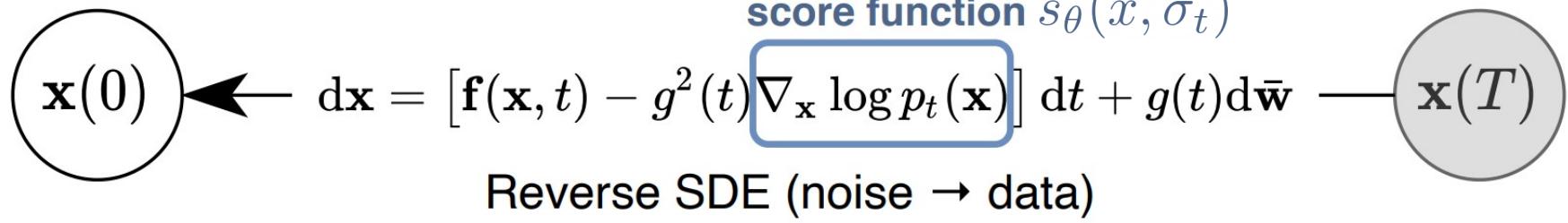
→ Train a model with arbitrary number forward steps but only sample from some of them in the generative process

# SDE-based Generative Models: A Unified Framework\*

Forward SDE (data → noise)



score function  $s_\theta(x, \sigma_t)$



Reverse SDE (noise → data)

# SDE-based Generative Models: A Unified Framework

## Training Objective (DSM)

$$\theta^* = \arg \min \mathbb{E}_{t \sim U(0, T)} \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{x(t)|x(0)} [\| s_\theta(x(t), t) - \nabla_{x(t)} \log p_{0t}(x(t)|x(0)) \|_2^2] \right\}$$

known Gaussian when  $f(x, t)$  if affine

## Discretizations

$$dx = f(x, t)dt + g(t)dw$$

SDE Form	Discrete Markov Chain	SDE Expression
Variance Exploding (VE) SDE (NCSN)	$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}$	$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw$
Variance Preserving (VP) SDE (DDPM)	$x_i = \sqrt{1 - \beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}$	$dx = \frac{1}{2} \beta(t) x dt + \sqrt{\beta(t)} dw$

# SDE-based Generative Models: A Unified Framework

**Model:** DDPM and SDE point of views

Score in score-based model is affine transformation of predicted noise in DDPM

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \boldsymbol{\varepsilon} \quad \text{Equivalent one step forward}$$

$$\begin{aligned} s_\theta(\mathbf{x}_t, t) &\approx \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{x}_0) && \text{Denoising score matching} \\ &= -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0}{1 - \bar{\alpha}_t} && \text{Gaussian assumption} \\ &= -\frac{\boldsymbol{\varepsilon}}{\sqrt{1 - \bar{\alpha}_t}} && \leftarrow \\ &\approx -\frac{\boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}} \end{aligned}$$

# Content

---

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

# SDE-based Generative Models: A Unified Framework

## Controllable Generation

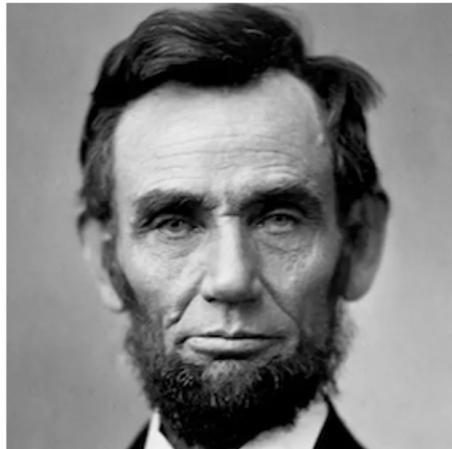
$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x|y)]dt + g(t)d\bar{w}$$

↓  
Bayesian

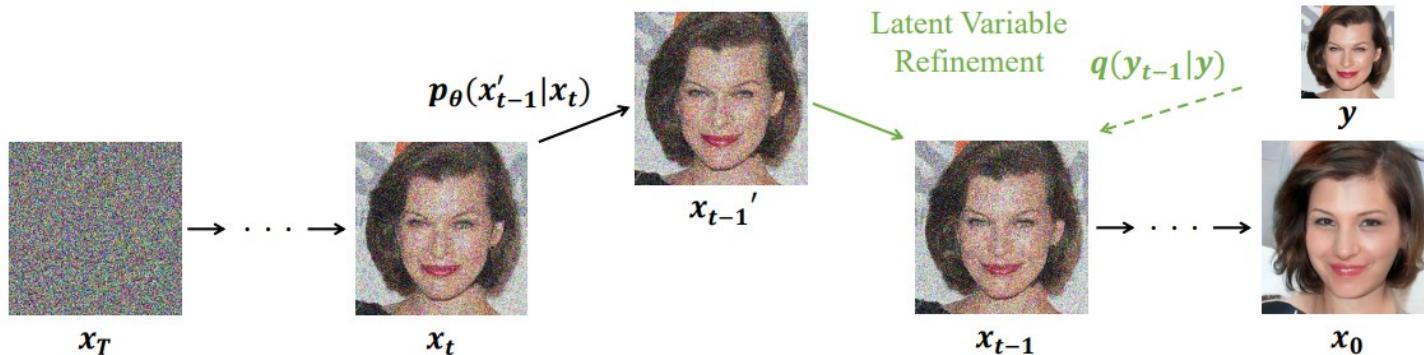
$$dx = \{f(x, t) - g(t)^2 [\nabla_x \log p_t(x) + \nabla_x \log p_t(y|x)]\}dt + g(t)d\bar{w}$$

unconditional model

time-dependent classifier(?)



# ILVR: Conditioning Method for DDPM\*



---

## Algorithm 1 Iterative Latent Variable Refinement

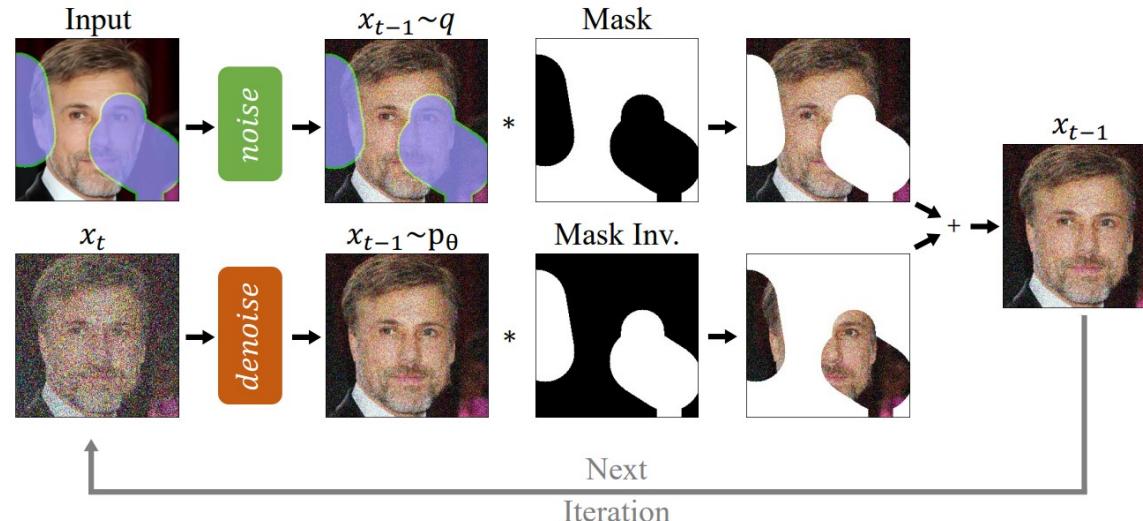
---

```
1: Input: Reference image  $y$ 
2: Output: Generated image  $x$ 
3:  $\phi_N(\cdot)$ : low-pass filter with scale N
4: Sample  $x_T \sim N(\mathbf{0}, \mathbf{I})$ 
5: for  $t = T, \dots, 1$  do
6:    $\mathbf{z} \sim N(\mathbf{0}, \mathbf{I})$ 
7:    $x'_{t-1} \sim p_\theta(x'_{t-1} | x_t)$        $\triangleright$  unconditional proposal
8:    $y_{t-1} \sim q(y_{t-1} | y)$              $\triangleright$  condition encoding
9:    $x_{t-1} \leftarrow \phi_N(y_{t-1}) + x'_{t-1} - \phi_N(x'_{t-1})$  xt-1 ← φN(yt-1) + x't-1 - φN(x't-1)  $\rightarrow x_{t-1} = x'_{t-1} + a(\phi_N(y_{t-1}) - \phi_N(x'_{t-1}))$ 
10: end for
11: return  $x_0$ 
```

---

# RePaint: Inpainting using Denoising Diffusion Probabilistic Models\*

Same idea but different downstream tasks from ILVR



**Algorithm 1** Inpainting using our RePaint approach.

```
1:  $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:   for  $u = 1, \dots, U$  do
4:      $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\epsilon = 0$ 
5:      $x_{t-1}^{\text{known}} = \sqrt{\alpha_t} x_0 + (1 - \bar{\alpha}_t) \epsilon$  unconditional
6:      $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $z = \mathbf{0}$ 
7:      $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z$ 
8:      $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$ 
9:     if  $u < U$  and  $t > 1$  then
10:       $x_t \sim \mathcal{N}(\sqrt{1 - \beta_{t-1}} x_{t-1}, \beta_{t-1} \mathbf{I})$ 
11:    end if
12:  end for
13: end for
14: return  $x_0$ 
```

# Diffusion Posterior Sampling for General Noisy Inverse Problems\*

general forward model  $\mathbf{y} = \mathcal{A}(\mathbf{x}_0) + \mathbf{n}, \quad \mathbf{y}, \mathbf{n} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{R}^d$

$$p(\mathbf{y}|\mathbf{x}_0) = \frac{1}{\sqrt{(2\pi)^n \sigma^{2n}}} \exp \left[ -\frac{\|\mathbf{y} - \mathcal{A}(\mathbf{x}_0)\|_2^2}{2\sigma^2} \right]$$

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)$$

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) \simeq s_{\theta^*}(\mathbf{x}_t, t) - \rho \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\hat{\mathbf{x}}_0)\|_2^2$$

---

## Algorithm 2 DPS - Gaussian [8]

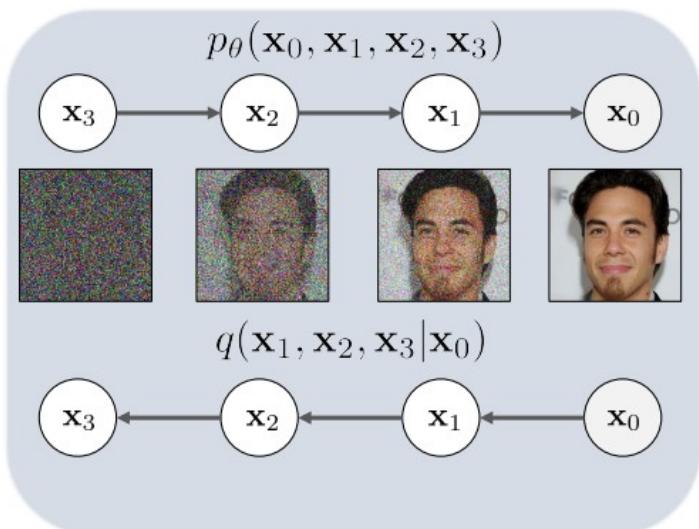
---

**Require:**  $N, \mathbf{y}, \{\zeta_i\}_{i=1}^N, \{\tilde{\sigma}_i\}_{i=1}^N$

- 1:  $x_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - 2: **for**  $i = N - 1$  **to** 0 **do**
  - 3:      $\hat{s} \leftarrow s_\theta(x_i, i)$
  - 4:      $\hat{x}_0 \leftarrow \frac{1}{\sqrt{\bar{\alpha}_i}}(x_i + \sqrt{1 - \bar{\alpha}_i} \hat{s})$
  - 5:      $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - 6:      $x'_{i-1} \leftarrow \frac{\sqrt{\alpha_i}(1 - \bar{\alpha}_{i-1})}{1 - \bar{\alpha}_i} x_i + \frac{\sqrt{\bar{\alpha}_{i-1}} \beta_i}{1 - \bar{\alpha}_i} \hat{x}_0 + \tilde{\sigma}_i z$
  - 7:      $x_{i-1} \leftarrow x'_{i-1} - \zeta_i \nabla_{x_i} \|\mathbf{y} - \mathcal{A}(\hat{x}_0)\|_2^2$
  - 8: **return**  $x_0$
-

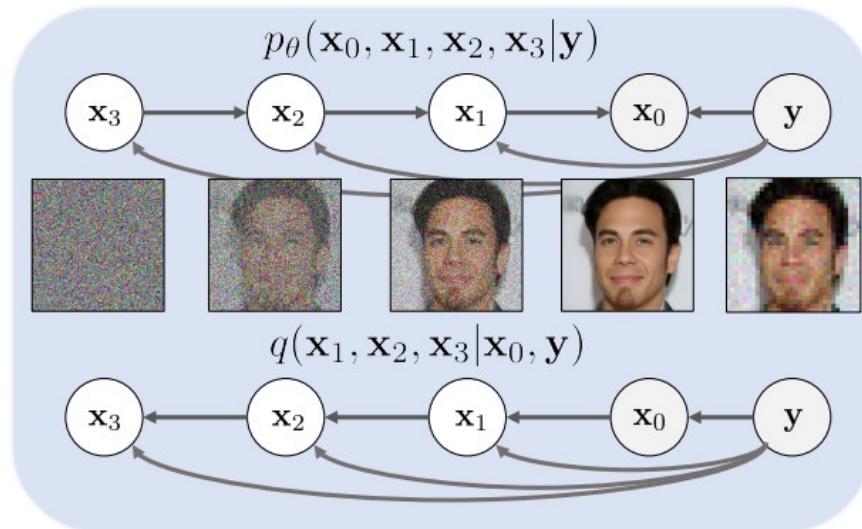
# Denoising Diffusion Restoration Models (DDRM)\*

An efficient, unsupervised posterior sampling method



Denoising Diffusion Probabilistic Models  
(Independent of inverse problem)

Use pre-trained  
models for linear  
inverse problems



Denoising Diffusion Restoration Models  
(Dependent on inverse problem)

# Denoising Diffusion Restoration Models (DDRM)\*

Linear inverse problem  $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z} \iff \bar{\mathbf{y}} = \bar{\mathbf{x}}_0 + \bar{\mathbf{z}}$      $q(\bar{\mathbf{y}}^{(i)} | \mathbf{x}_0) = \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_y^2 / s_i^2)$

$$\text{SVD } \mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^\top \iff \bar{\mathbf{x}}_t = \mathbf{V}^T \mathbf{x}_t \\ \bar{\mathbf{y}} = \Sigma^\dagger \mathbf{U}^T \mathbf{y}$$

$q^{(T)}(\bar{\mathbf{x}}_T^{(i)}   \mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_y^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)}, \sigma_T^2) & \text{if } s_i = 0 \end{cases}$ <b>forward</b>	$q^{(t)}(\bar{\mathbf{x}}_t^{(i)}   \mathbf{x}_{t+1}, \mathbf{x}_0, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_{t+1}}, \eta^2 \sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_0^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_0^{(i)}}{\sigma_y / s_i}, \eta^2 \sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_y}{s_i} \\ \mathcal{N}((1 - \eta_b) \bar{\mathbf{x}}_0^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_y^2}{s_i^2} \eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_y}{s_i} \end{cases}$ <b>reverse</b>
--	---

$$p_\theta^{(T)}(\bar{\mathbf{x}}_T^{(i)} | \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{y}}^{(i)}, \sigma_T^2 - \frac{\sigma_y^2}{s_i^2}) & \text{if } s_i > 0 \\ \mathcal{N}(0, \sigma_T^2) & \text{if } s_i = 0 \end{cases} \quad \text{singular values } s_1 \geq s_2 \geq \dots \geq s_m$$

$$\text{DDRM} \quad p_\theta^{(t)}(\bar{\mathbf{x}}_t^{(i)} | \mathbf{x}_{t+1}, \mathbf{y}) = \begin{cases} \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{x}}_{t+1}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_{t+1}}, \eta^2 \sigma_t^2) & \text{if } s_i = 0 \\ \mathcal{N}(\bar{\mathbf{x}}_{\theta,t}^{(i)} + \sqrt{1 - \eta^2} \sigma_t \frac{\bar{\mathbf{y}}^{(i)} - \bar{\mathbf{x}}_{\theta,t}^{(i)}}{\sigma_y / s_i}, \eta^2 \sigma_t^2) & \text{if } \sigma_t < \frac{\sigma_y}{s_i} \\ \mathcal{N}((1 - \eta_b) \bar{\mathbf{x}}_{\theta,t}^{(i)} + \eta_b \bar{\mathbf{y}}^{(i)}, \sigma_t^2 - \frac{\sigma_y^2}{s_i^2} \eta_b^2) & \text{if } \sigma_t \geq \frac{\sigma_y}{s_i} \end{cases}$$

y null-space      final steps      generative part

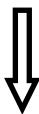
# Denoising Diffusion Models for Plug-and-Play Image Restoration

## HQS algorithm

$$\begin{cases} \mathbf{z}_k = \arg \min_{\mathbf{z}} \underbrace{\frac{1}{2(\sqrt{\lambda/\mu})^2} \|\mathbf{z} - \mathbf{x}_k\|^2}_{\text{consistency}} + \underbrace{\mathcal{P}(\mathbf{z})}_{\text{prior}} \\ \mathbf{x}_{k-1} = \arg \min_{\mathbf{x}} \underbrace{\|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2}_{\text{condition}} + \underbrace{\mu\sigma_n^2 \|\mathbf{x} - \mathbf{z}_k\|^2}_{\text{consistency}} \end{cases}$$

$$\begin{cases} \mathbf{z}_0 = \arg \min_{\mathbf{z}} \frac{1}{2\bar{\sigma}_t^2} \|\mathbf{z} - \mathbf{x}_t\|^2 + \mathcal{P}(\mathbf{z}) \\ \mathbf{x}_0 = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \rho_t \|\mathbf{x} - \mathbf{z}_0\|^2 \\ \mathbf{x}_{t-1} \leftarrow \mathbf{x}_0. \end{cases}$$

unconditional




---

## Algorithm 1 DiffPIR

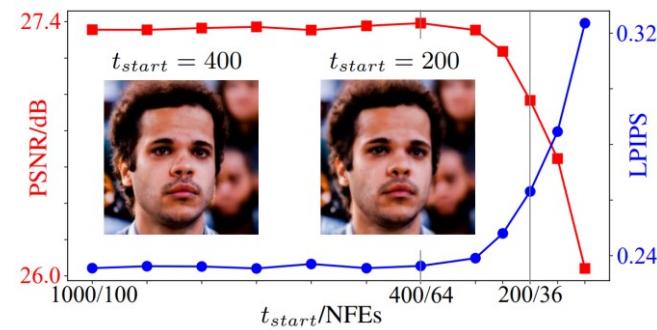
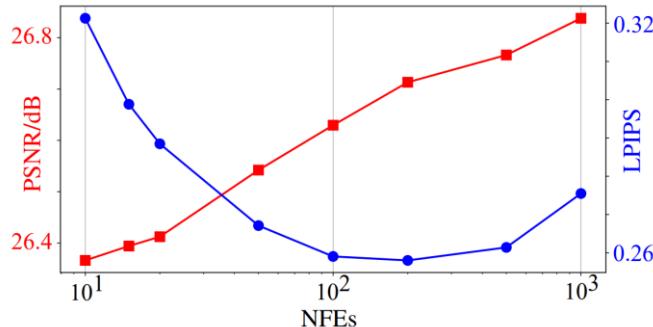
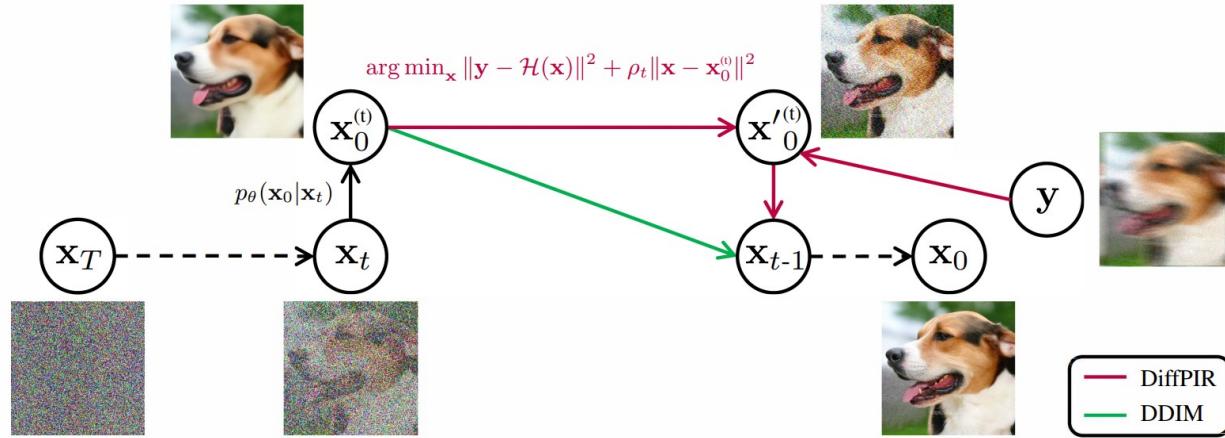
---

**Require:**  $\mathbf{s}_\theta, T, \mathbf{y}, \sigma_n, \{\bar{\sigma}_t\}_{t=1}^T, \zeta, \lambda$

- 1: Initialize  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ , pre-calculate  $\rho_t \triangleq \lambda\sigma_n^2/\bar{\sigma}_t^2$ .
  - 2: **for**  $t = T$  **to** 1 **do**
  - 3:      $\hat{\mathbf{z}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t + (1 - \bar{\alpha}_t)\mathbf{s}_\theta(\mathbf{x}_t, t))$  // Predict  $\hat{\mathbf{z}}_0$  with score model as denoisor
  - 4:      $\hat{\mathbf{x}}_0 = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathcal{H}(\mathbf{x})\|^2 + \rho_t \|\mathbf{x} - \hat{\mathbf{z}}_0\|^2$  // Solving data proximal subproblem
  - 5:      $\hat{\epsilon} = \frac{1}{\sqrt{1 - \bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \hat{\mathbf{x}}_0)$  // Calculate effective  $\hat{\epsilon}(\mathbf{x}_t, \mathbf{y})$
  - 6:      $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - 7:      $\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \hat{\mathbf{x}}_0 + \sqrt{1 - \bar{\alpha}_{t-1}} (\sqrt{1 - \zeta} \hat{\epsilon} + \sqrt{\zeta} \epsilon_t)$  // Finish one step reverse diffusion sampling
  - 8: **end for**
  - 9: **return**  $\mathbf{x}_0$
- 

Approximately  $\hat{\mathbf{x}}_0 \approx \hat{\mathbf{z}}_0 - \frac{\bar{\sigma}_t^2}{2\lambda\sigma_n^2} \nabla_{\mathbf{z}_0} \|\mathbf{y} - \mathcal{H}(\mathbf{z}_0)\|^2$ .

# Denoising Diffusion Models for Plug-and-Play Image Restoration\*



# Diffusion Model Based Posterior Sampling for Noisy Linear Inverse Problems\*

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) &\simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t) \\ &= \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A}^T \left( \sigma^2 \mathbf{I} + \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \mathbf{A} \mathbf{A}^T \right)^{-1} \left( \mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A} \mathbf{x}_t \right) \end{aligned}$$

**A itself is row-orthogonal**

$$[\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)]_m = \frac{\mathbf{a}_m^T \left( \mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{A} \mathbf{x}_t \right)}{\sigma^2 \sqrt{\bar{\alpha}_t} + \frac{1 - \bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \|\mathbf{a}_m\|_2^2}$$

**efficient computation via SVD**

$$\begin{aligned} \nabla_{\mathbf{x}_t} \log p(\mathbf{y} \mid \mathbf{x}_t) &\simeq \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t) \\ &= \frac{1}{\sqrt{\bar{\alpha}_t}} \mathbf{V} \boldsymbol{\Sigma} \left( \sigma^2 \mathbf{I} + \frac{1 - \bar{\alpha}_t}{\bar{\alpha}_t} \boldsymbol{\Sigma}^2 \right)^{-1} \left( \mathbf{U}^T \mathbf{y} - \frac{1}{\sqrt{\bar{\alpha}_t}} \boldsymbol{\Sigma} \mathbf{V}^T \mathbf{x}_t \right), \end{aligned} \tag{12}$$

---

**Algorithm 1:** DMPS: DM based posterior sampling

---

**Input:**  $\mathbf{y}, \mathbf{A}, \sigma^2, \{\tilde{\sigma}_t\}_{t=1}^T, \lambda$

**Initialization:**  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

- 1 **for**  $t = T$  **to** 1 **do**
- 2     Draw  $\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 3      $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \mathbf{x}_t - \frac{1 - \bar{\alpha}_t}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{s}_{\theta}(\mathbf{x}_t, t) \right) + \tilde{\sigma}_t \mathbf{z}_t$
- 4     Compute  $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$  as (12)
- 5      $\mathbf{x}_{t-1} = \mathbf{x}_{t-1} + \lambda \frac{1 - \bar{\alpha}_t}{\sqrt{\bar{\alpha}_t}} \nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{y} \mid \mathbf{x}_t)$

---

**Output:**  $\mathbf{x}_0$

---

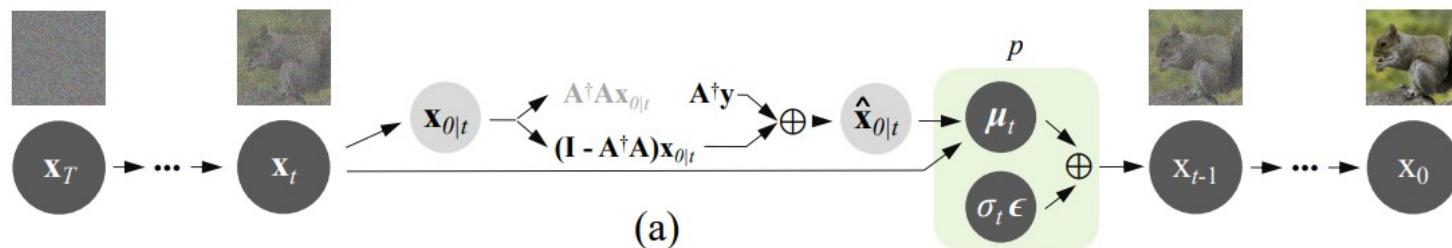
# Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model\*

**Decouple**     $\mathbf{x} \equiv \underbrace{\mathbf{A}^\dagger \mathbf{A} \mathbf{x}}_{\text{range-space of } \mathbf{A}} + \underbrace{(\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}}_{\text{null-space of } \mathbf{A}}$

*Consistency* :     $\mathbf{A}\hat{\mathbf{x}} \equiv \mathbf{y}$ ,        *Realness* :     $\hat{\mathbf{x}} \sim q(\mathbf{x})$

**Reconstruction**     $\hat{\mathbf{x}} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \bar{\mathbf{x}}$         find a proper  $\bar{\mathbf{x}}$  that makes the null-space term is in harmony with the range-space term

**Diffusion Models**     $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}_{0|t}$



# Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model\*

---

## Algorithm 1 Sampling of DDNM

---

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{x}_{0|t} = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \mathcal{Z}_\theta(\mathbf{x}_t, t) \sqrt{1 - \bar{\alpha}_t})$ 
4:    $\hat{\mathbf{x}}_{0|t} = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) \mathbf{x}_{0|t}$ 
5:    $\mathbf{x}_{t-1} \sim p(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_{0|t})$ 
6: return  $\mathbf{x}_0$ 

```

---

## Algorithm 2 Sampling of DDNM<sup>+</sup>

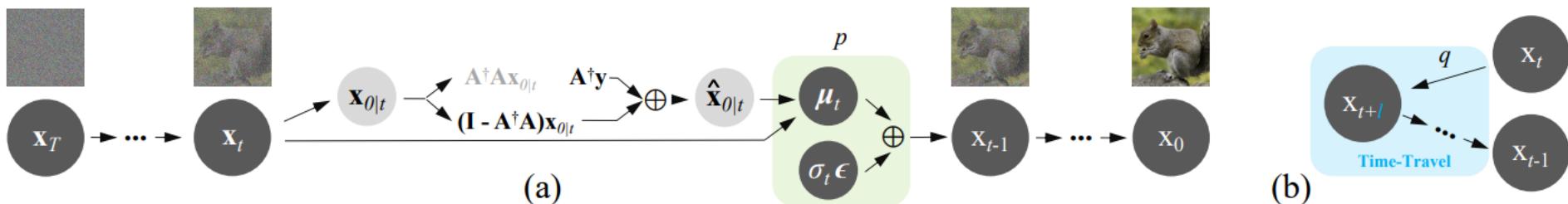
---

```

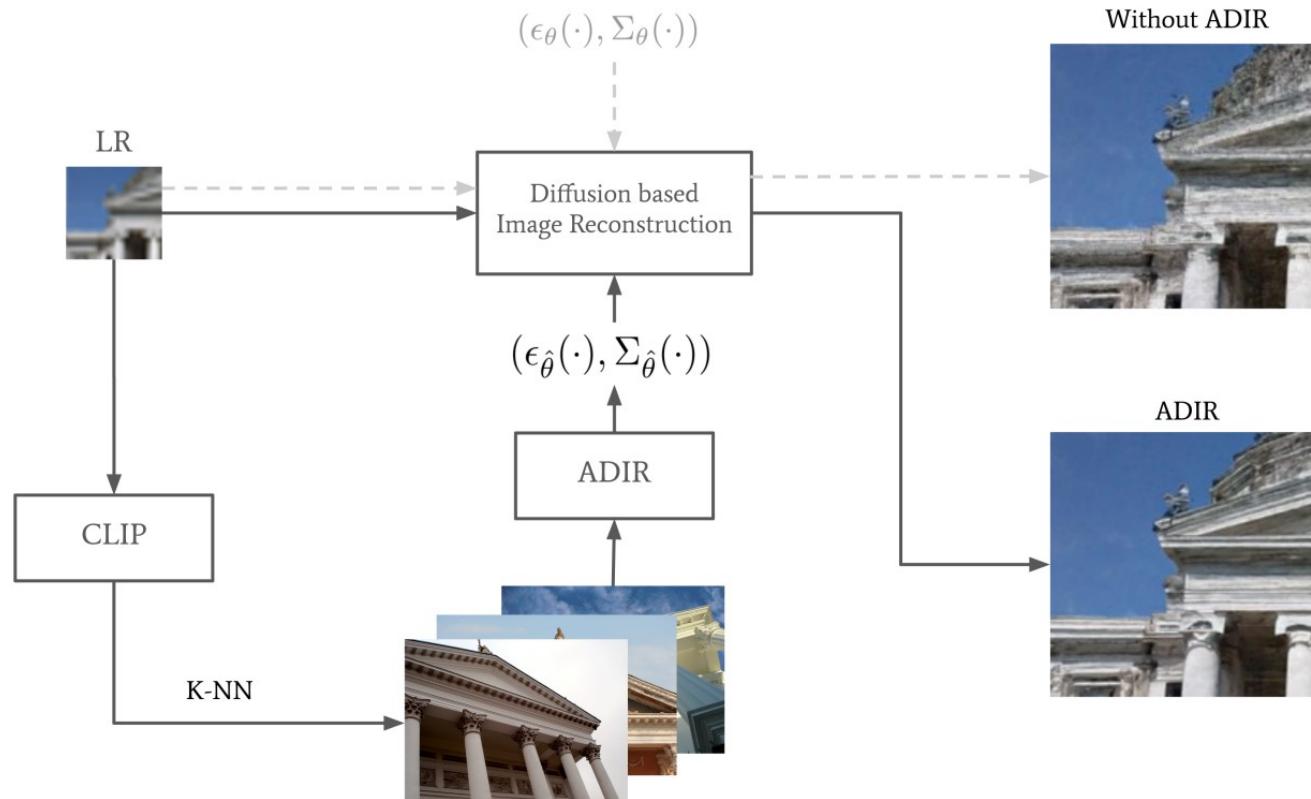
1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $L = \min\{T - t, l\}$ 
4:    $\mathbf{x}_{t+L} \sim q(\mathbf{x}_{t+L} | \mathbf{x}_t)$ 
5:   for  $j = L, \dots, 0$  do
6:      $\mathbf{x}_{0|t+j} = \frac{1}{\sqrt{\bar{\alpha}_{t+j}}} (\mathbf{x}_{t+j} - \mathcal{Z}_\theta(\mathbf{x}_{t+j}, t + j) \sqrt{1 - \bar{\alpha}_{t+j}})$ 
7:      $\hat{\mathbf{x}}_{0|t+j} = \mathbf{x}_{0|t+j} - \Sigma_{t+j} \mathbf{A}^\dagger (\mathbf{A} \mathbf{x}_{0|t+j} - \mathbf{y})$ 
8:      $\mathbf{x}_{t+j-1} \sim \hat{p}(\mathbf{x}_{t+j-1} | \mathbf{x}_{t+j}, \hat{\mathbf{x}}_{0|t+j})$ 
9: return  $\mathbf{x}_0$ 

```

---



# ADIR: Adaptive Diffusion for Image Reconstruction\*



\*[2212.03221] ADIR: Adaptive Diffusion for Image Reconstruction ([arxiv.org/](https://arxiv.org/))

ADIR: Adaptive Diffusion for Image Reconstruction ([shadyabh.github.io](https://shadyabh.github.io))

# ADIR: Adaptive Diffusion for Image Reconstruction\*

---

**Algorithm 1** Proposed guided diffusion sampling for image reconstruction given a diffusion model  $(\epsilon_\theta(\cdot), \Sigma_\theta(\cdot))$ , and a guidance scale  $s$

---

**Require:**  $(\epsilon_\theta(\cdot), \Sigma_\theta(\cdot)), \mathbf{y}, s$

- 1:  $\mathbf{x}_T \leftarrow$  sample from  $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$
  - 2: **for all**  $t$  from  $T$  to 1 **do**
  - 3:    $\hat{\epsilon}, \hat{\Sigma} \leftarrow \epsilon_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)$
  - 4:    $\hat{\mu} \leftarrow \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon})$
  - 5:    $\mathbf{y}_t \leftarrow \sqrt{\bar{\alpha}_t} \mathbf{y} + \sqrt{1-\bar{\alpha}_t} \mathbf{A} \hat{\epsilon}$
  - 6:    $\mathbf{g} \leftarrow -2\mathbf{A}^T (\mathbf{A} \hat{\mu} - \mathbf{y}_t)$
  - 7:    $\mathbf{x}_{t-1} \leftarrow$  sample from  $\mathcal{N}(\hat{\mu} + s \hat{\Sigma} \mathbf{g}, \hat{\Sigma})$
  - 8: **end for**
  - 9: **return**  $\mathbf{x}_0$
- 

$$\hat{\mu}_\theta = \mu_\theta + a \Sigma_\theta \nabla_x \log p_\phi(y | \mathbf{x}_{t-1})$$

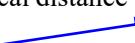
$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) \simeq -\rho \nabla_{\mathbf{x}_t} \|\mathbf{y} - \mathcal{A}(\mathbf{x}_0)\|_\Lambda^2, \quad [\Lambda]_{ii} \triangleq 1/2\mathbf{y}_j$$

$$\mathbf{g} \approx -2\mathbf{A}^T (\mathbf{A} \mathbf{x}_t - \mathbf{y}_t) |_{\mathbf{x}_t=\mu_\theta}$$

**Adaptation scheme:**

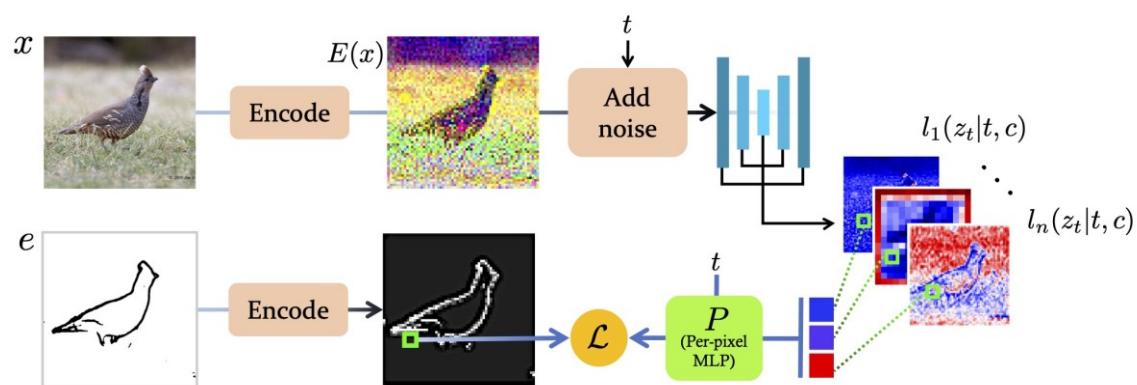
retrieve K images to fine-tuning the diffusion model

spherical distance between embeddings

$$\begin{aligned} \{\mathbf{z}_k\}_{k=1}^K &= \{\mathbf{z}_1, \dots, \mathbf{z}_K | \phi_\xi(\mathbf{z}_1, \mathbf{y}) \leq \dots \leq \phi_\xi(\mathbf{z}_K, \mathbf{y}) \\ &\leq \phi_\xi(\mathbf{z}, \mathbf{y}), \forall \mathbf{z} \in \mathcal{D}_{IA} \setminus \{\mathbf{z}_1, \dots, \mathbf{z}_K\}\}, \end{aligned}$$


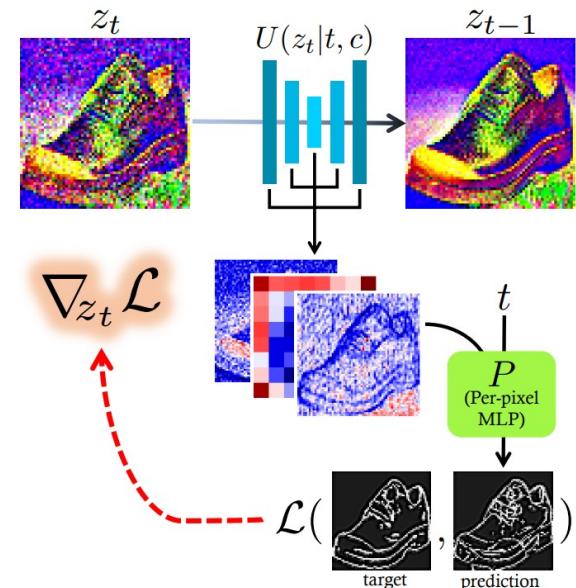
# Sketch-Guided Text-to-Image Diffusion Models\*

(1) Train Per-pixel MLP for sketch-image spatial alignment



Update similar to DPS:  $\tilde{z}_{t-1} = z_{t-1} - \alpha \cdot \nabla_{z_t} \mathcal{L}$

(2) Evaluate the anti-gradient as edges-guidance

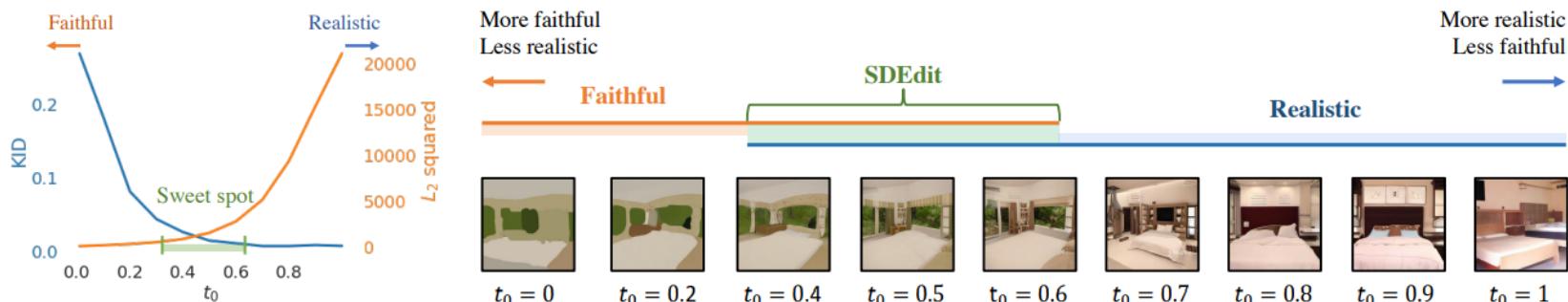
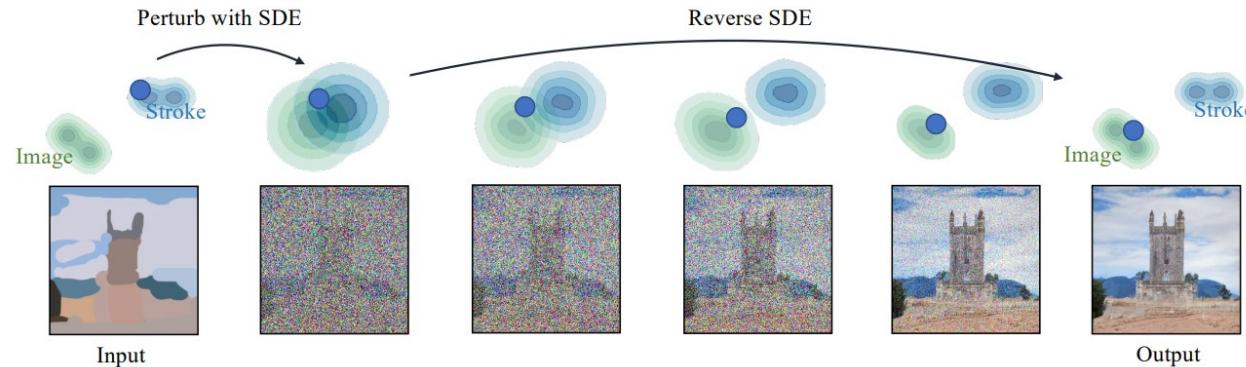


# Content

---

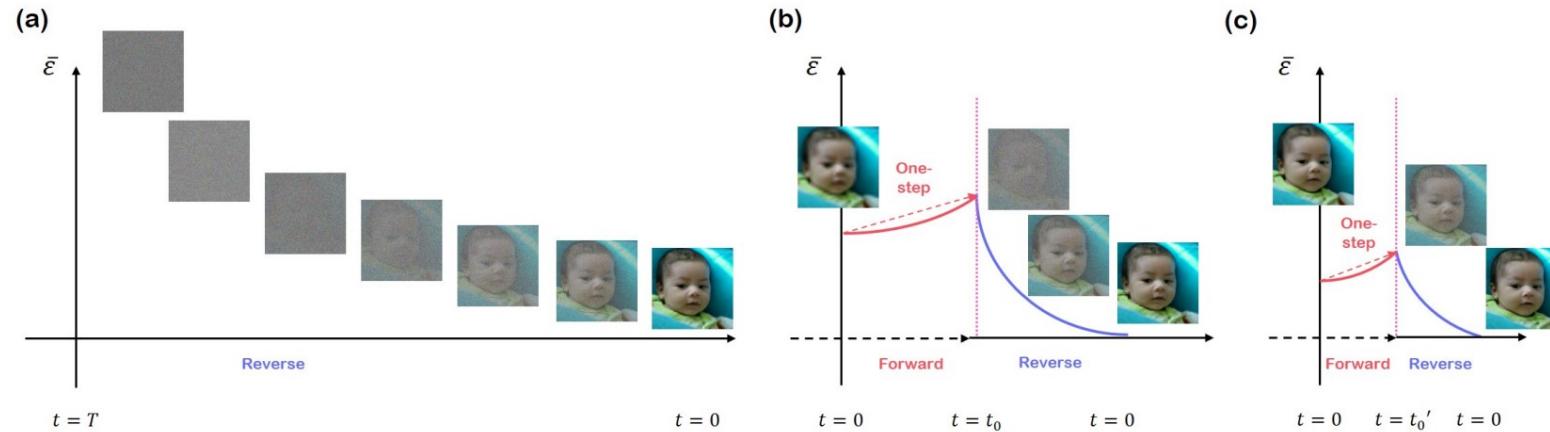
- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

# SDEdit: Guided Image Synthesis and Editing with SDE\*



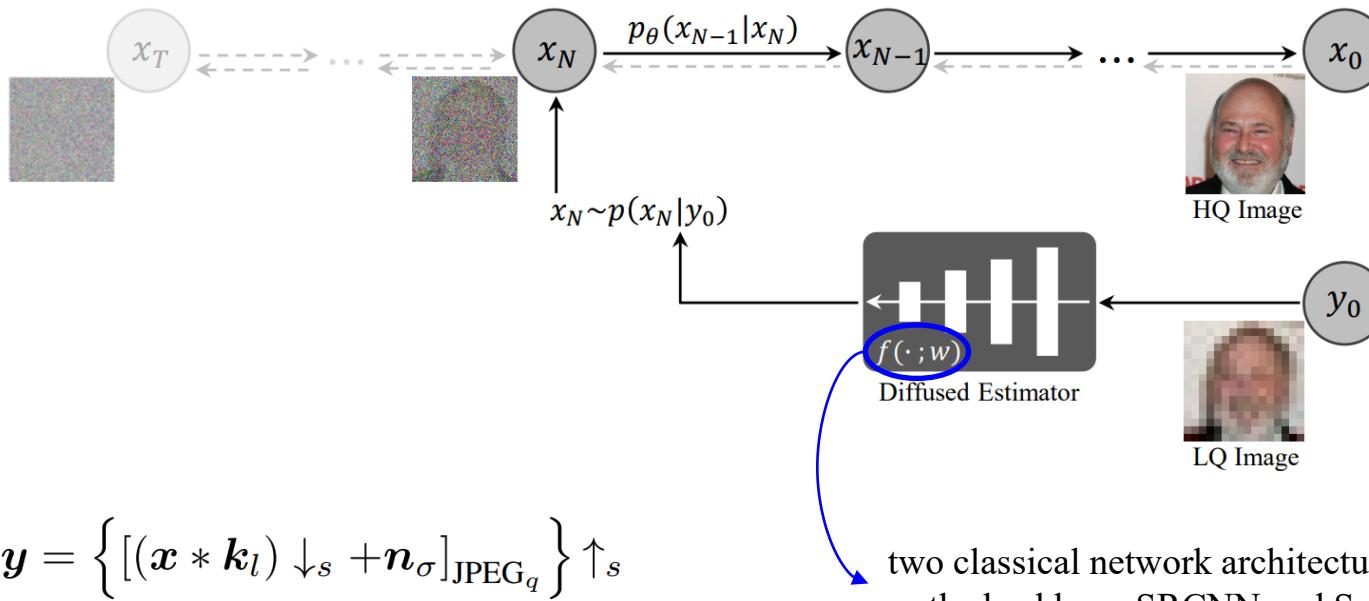
# Come-Closer-Diffuse-Faster: Accelerating Conditional Diffusion Models for Inverse Problems through Stochastic Contraction\*

Same idea but different downstream tasks: super-resolution (SR), inpainting, and MRI reconstruction



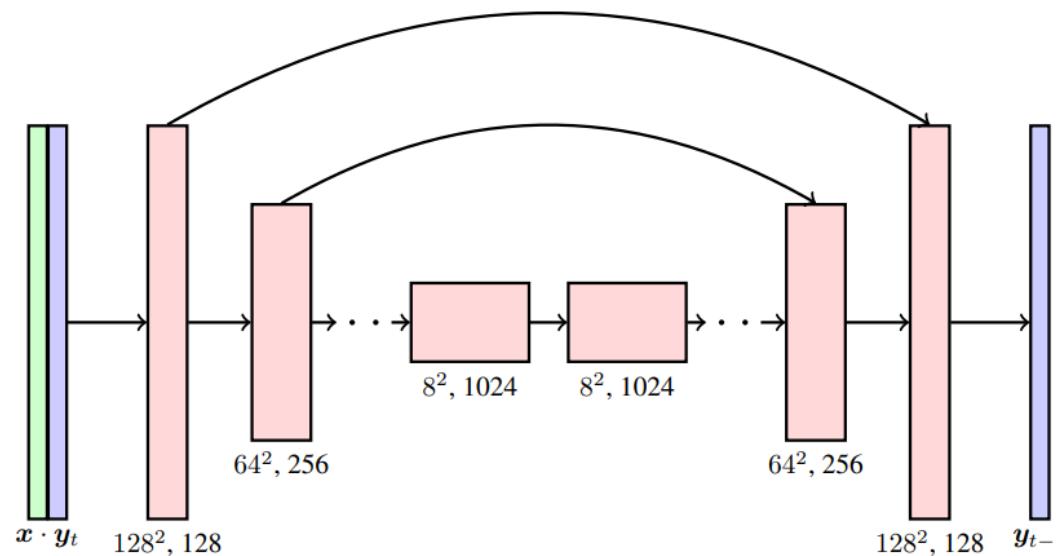
# DiffFace: Blind Face Restoration with Diffused Error Contraction\*

Same idea but different downstream task: Blind face (easier) restoration



# Image Super-Resolution via Iterative Refinement\*

The condition is concatenated with  $y_t$  along the channel dimension (cascaded)



Same author also proposed palette for multi-tasks<sup>†</sup>, same architecture used for cascaded diffusion<sup>‡</sup>

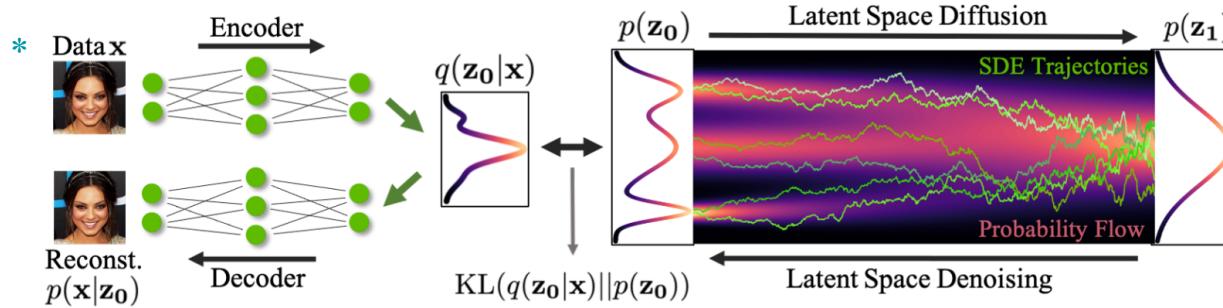
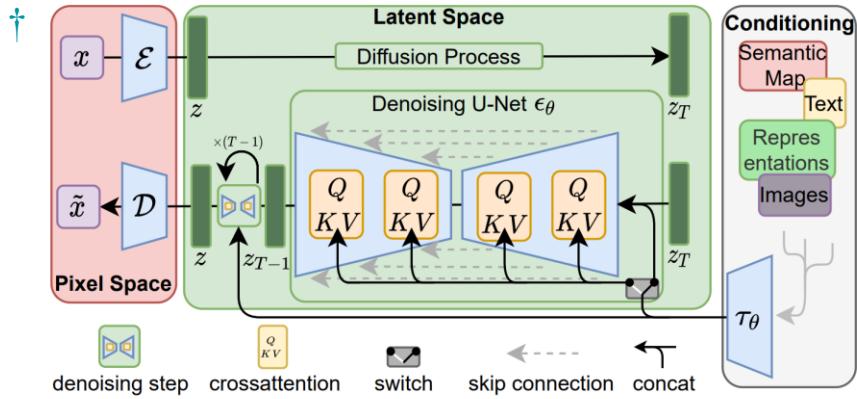
\*[2104.07636] Image Super-Resolution via Iterative Refinement ([arxiv.org/](https://arxiv.org/))

†[2111.05826] Palette: Image-to-Image Diffusion Models ([arxiv.org/](https://arxiv.org/))

‡[2106.15282] Cascaded Diffusion Models for High Fidelity Image Generation ([arxiv.org/](https://arxiv.org/))

# Score-based Generative Modeling in Latent Space\*

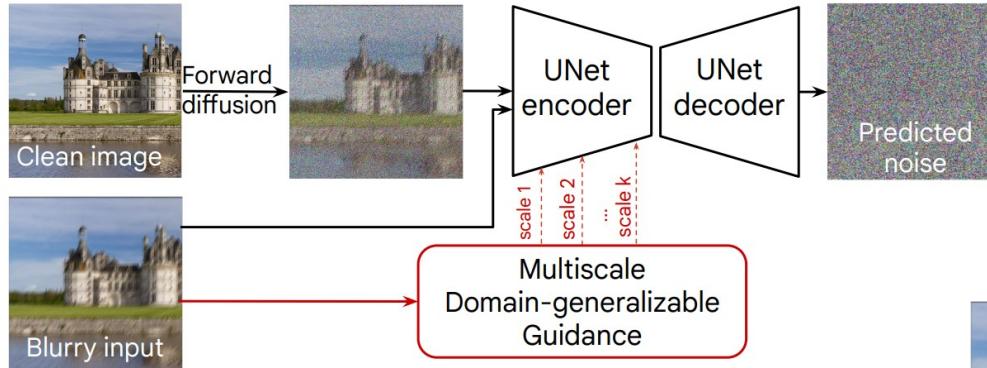
Faster diffusion  
in latent space



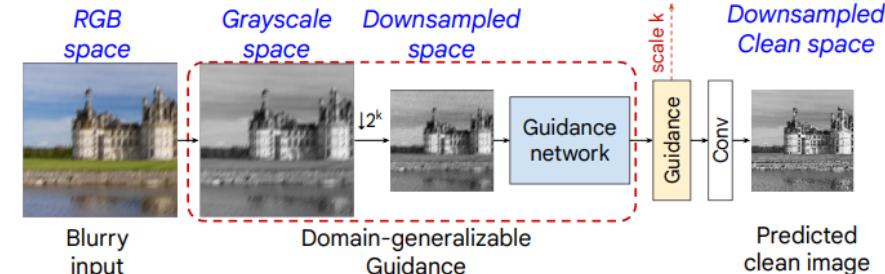
†[2112.10752] High-Resolution Image Synthesis with Latent Diffusion Models (arxiv.org)

\*[2106.05931] Score-based Generative Modeling in Latent Space (arxiv.org)

# Image Deblurring with Domain Generalizable Diffusion Models\*



multiscale domain-generalizable representation that removes domain-specific information while preserving the underlying image structure



# Content

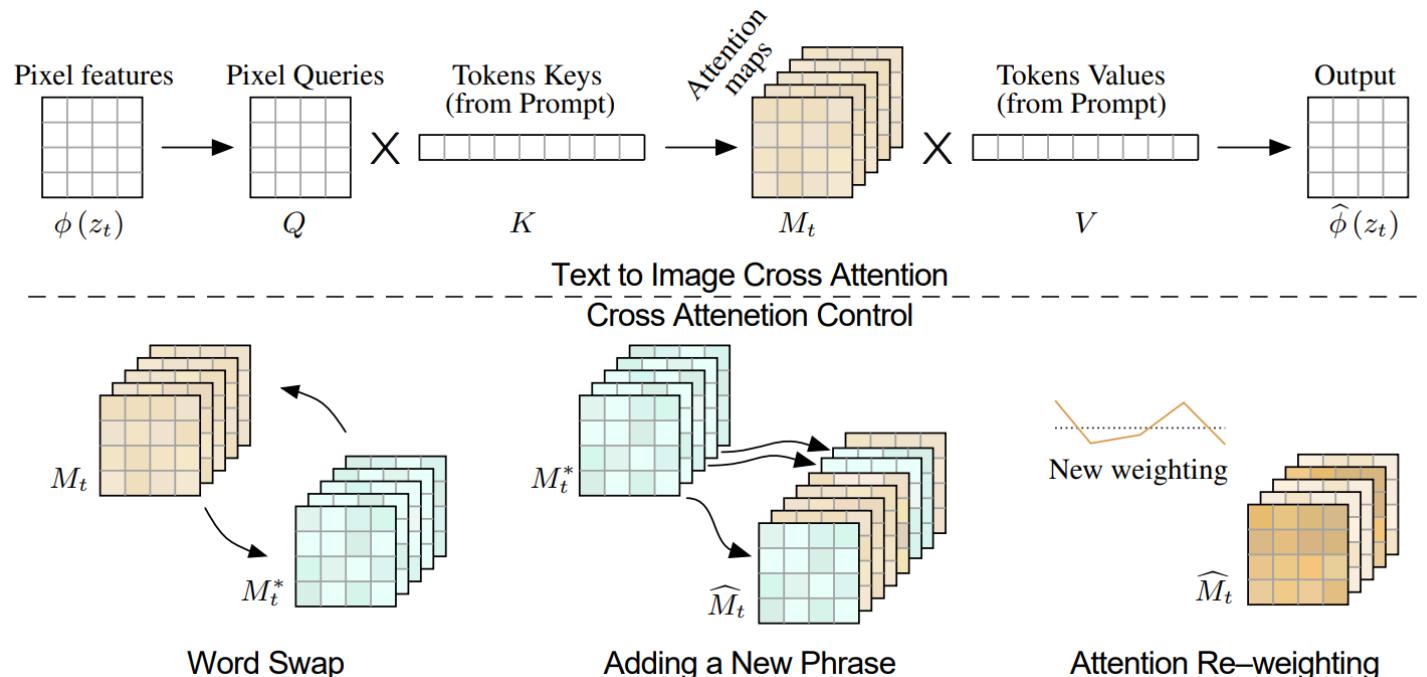
---

- Preliminaries
- Sampling from the Posterior
- Conditional Models
- Cross-attention Control

# Prompt-to-Prompt Image Editing with Cross Attention Control\*

$$M = \text{Softmax} \left( \frac{QK^T}{\sqrt{d}} \right)$$

we can inject the attention maps  $M$  that were obtained from the generation with the original prompt  $P$ , into a second generation with the modified prompt  $P^*$



# Prompt-to-Prompt Image Editing with Cross Attention Control\*

---

**Algorithm 1:** Prompt-to-Prompt image editing

---

```
1 Input: A source prompt  $\mathcal{P}$ , a target prompt  $\mathcal{P}^*$ , and a random seed  $s$ .  
2 Output: A source image  $x_{src}$  and an edited image  $x_{dst}$ .  
3  $z_T \sim N(0, I)$  a unit Gaussian random variable with random seed  $s$ ;  
4  $z_T^* \leftarrow z_T$ ;  
5 for  $t = T, T - 1, \dots, 1$  do  
6    $z_{t-1}, M_t \leftarrow DM(z_t, \mathcal{P}, t, s)$ ;  
7    $M_t^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s)$ ;  
8    $\widehat{M}_t \leftarrow Edit(M_t, M_t^*, t)$ ;  
9    $z_{t-1}^* \leftarrow DM(z_t^*, \mathcal{P}^*, t, s_t) \{M \leftarrow \widehat{M}_t\}$ ;  
10 end  
11 Return  $(z_0, z_0^*)$ 
```

---

**Word Swap**

$$Edit(M_t, M_t^*, t) := \begin{cases} M_t^* & \text{if } t < \tau \\ M_t & \text{otherwise} \end{cases}$$

**Adding a New Phrase**

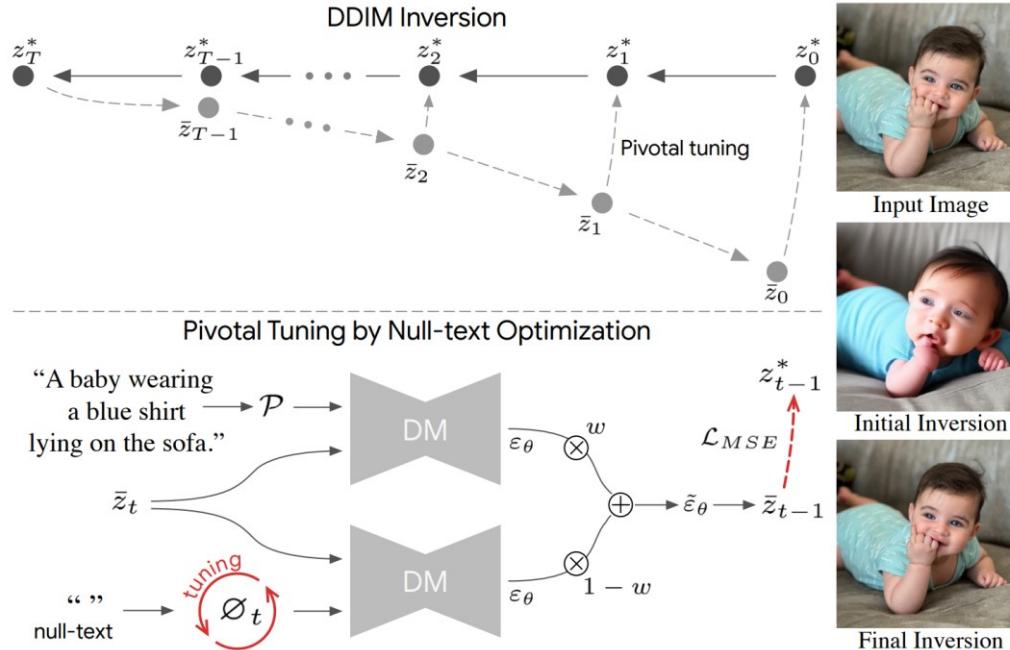
$$(Edit(M_t, M_t^*, t))_{i,j} := \begin{cases} (M_t^*)_{i,j} & \text{if } A(j) = None \\ (M_t)_{i,A(j)} & \text{otherwise.} \end{cases}$$

$i$  :pixel value;  $j$  :text token

**Attention Re-weighting**

$$(Edit(M_t, M_t^*, t))_{i,j} := \begin{cases} c \cdot (M_t)_{i,j} & \text{if } j = j^* \\ (M_t)_{i,j} & \text{otherwise} \end{cases}$$

# Null-text Inversion for Editing Real Images using Guided Diffusion Models\*




---

## Algorithm 1: Null-text inversion

---

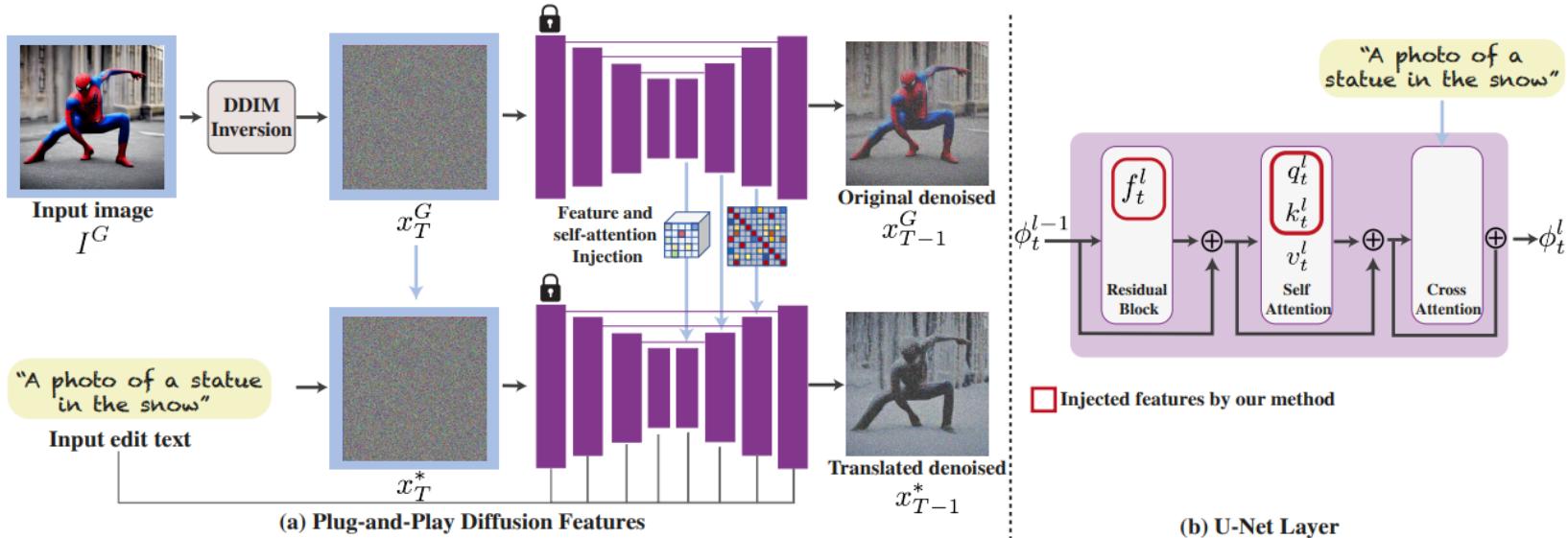
```

1 Input: A source prompt embedding  $\mathcal{C} = \psi(\mathcal{P})$  and input image  $\mathcal{I}$ .
2 Output: Noise vector  $z_T$  and optimized embeddings  $\{\emptyset_t\}_{t=1}^T$ .
3 Set guidance scale  $w = 1$ ;
4 Compute the intermediate results  $z_T^*, \dots, z_0^*$  using DDIM inversion over  $\mathcal{I}$ ;
5 Set guidance scale  $w = 7.5$ ;
6 Initialize  $\bar{z}_T \leftarrow z_T^*, \emptyset_T \leftarrow \psi(\text{""})$ ;
7 for  $t = T, T - 1, \dots, 1$  do
8   for  $j = 0, \dots, N - 1$  do
9      $\emptyset_t \leftarrow \emptyset_t - \eta \nabla_{\emptyset} \|\bar{z}_{t-1}^* - z_{t-1}(\bar{z}_t, \emptyset_t, \mathcal{C})\|_2^2$ 
10   end
11   Set  $\bar{z}_{t-1} \leftarrow z_{t-1}(\bar{z}_t, \emptyset_t, \mathcal{C}), \emptyset_{t-1} \leftarrow \emptyset_t$ ;
12 end
13 Return  $\bar{z}_T, \{\emptyset_t\}_{t=1}^T$ 

```

---

# Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation\*



# Plug-and-Play Diffusion Features for Text-Driven Image-to-Image Translation\*

---

## Algorithm 1 Plug-and-Play Diffusion Features

---

**Inputs:**

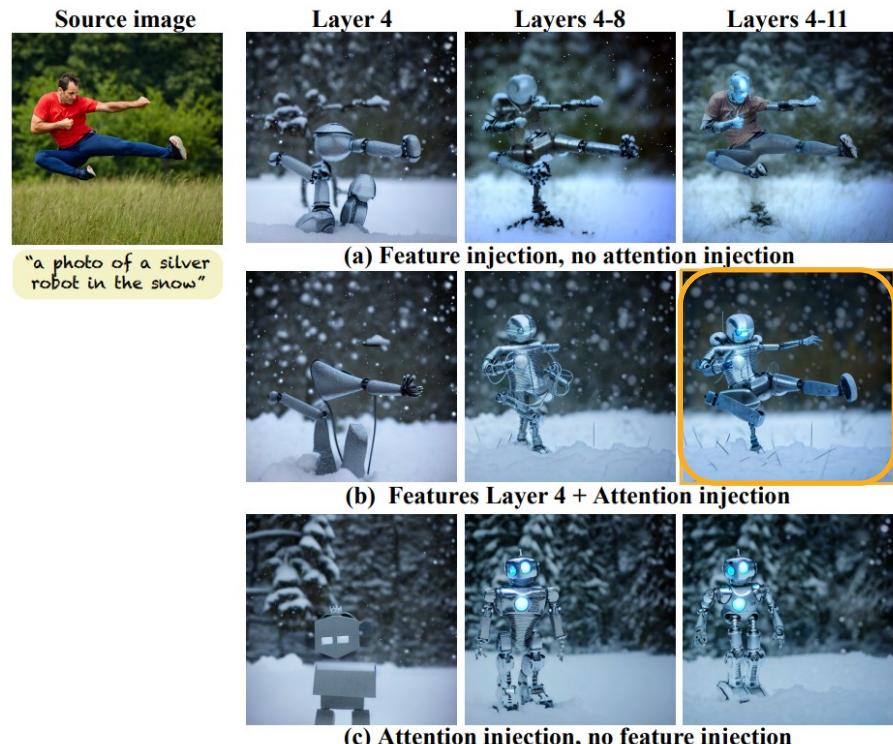
$I^G$  ▷ real guidance image

$P$  ▷ target text prompt

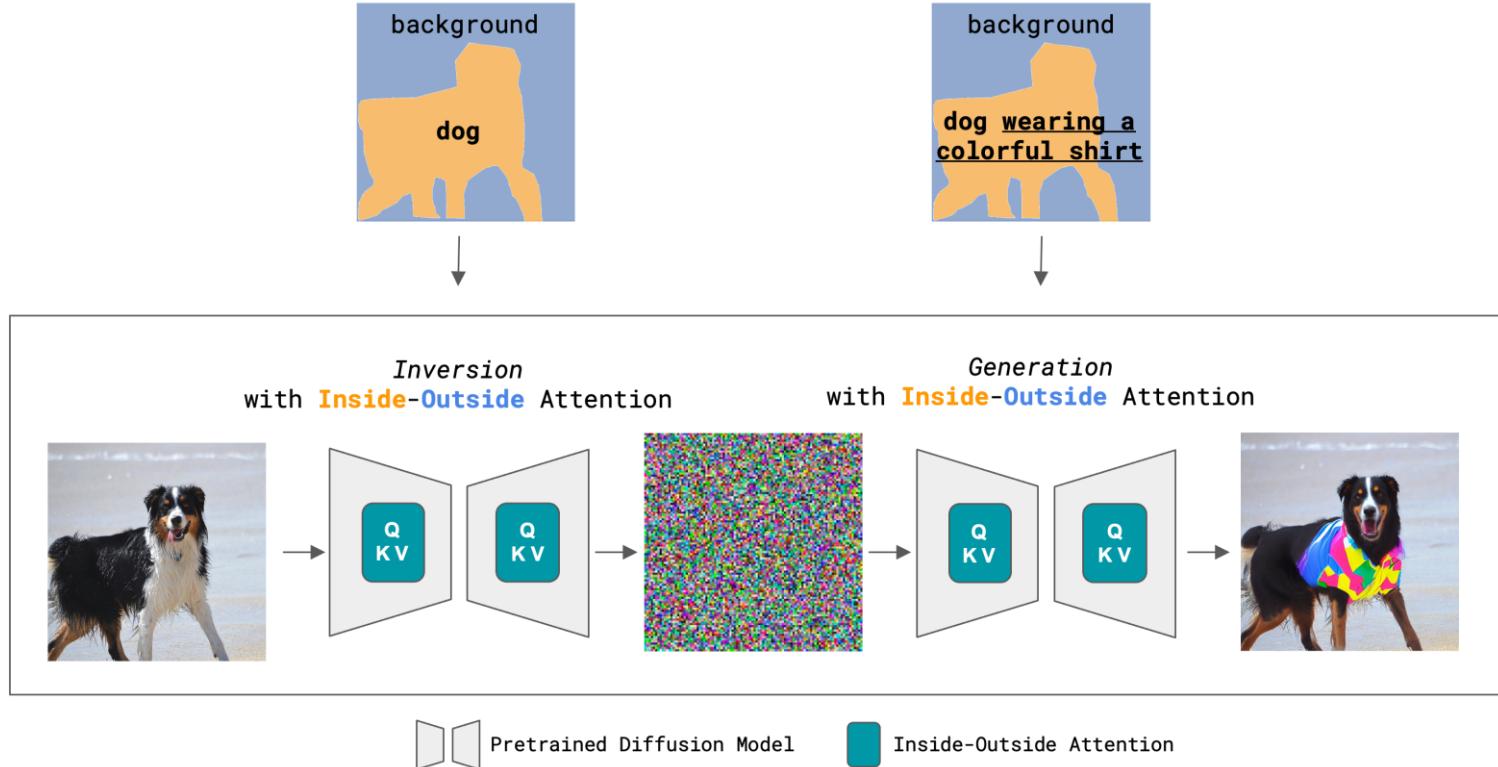
$\tau_f, \tau_A$  ▷ injection thresholds

```
 $x_T^G \leftarrow \text{DDIM-inv}(I^G)$ 
 $x_T^* \leftarrow x_T^G$                                 ▷ Starting from same seed
for  $t \leftarrow T \dots 1$  do
     $z_{t-1}^G, f_t^4, \{A_t^l\} \leftarrow \epsilon_\theta(x_t^G, \emptyset, t)$ 
     $x_{t-1}^G \leftarrow \text{DDIM-samp}(x_t^G, z_{t-1}^G)$ 
    if  $t > \tau_f$  then  $f_t^{*4} \leftarrow f_t^4$  else  $f_t^{*4} \leftarrow \emptyset$ 
    if  $t > \tau_A$  then  $A_t^{*l} \leftarrow A_t^l$  else  $A_t^{*l} \leftarrow \emptyset$ 
     $z_{t-1}^* \leftarrow \hat{\epsilon}_\theta(x_t^*, P, t ; f_t^{*4}, \{A_t^{*l}\})$ 
     $x_{t-1}^* \leftarrow \text{DDIM-samp}(x_t^*, z_{t-1}^*)$ 
end for
Output:  $I^* \leftarrow x_0^*$ 
```

---



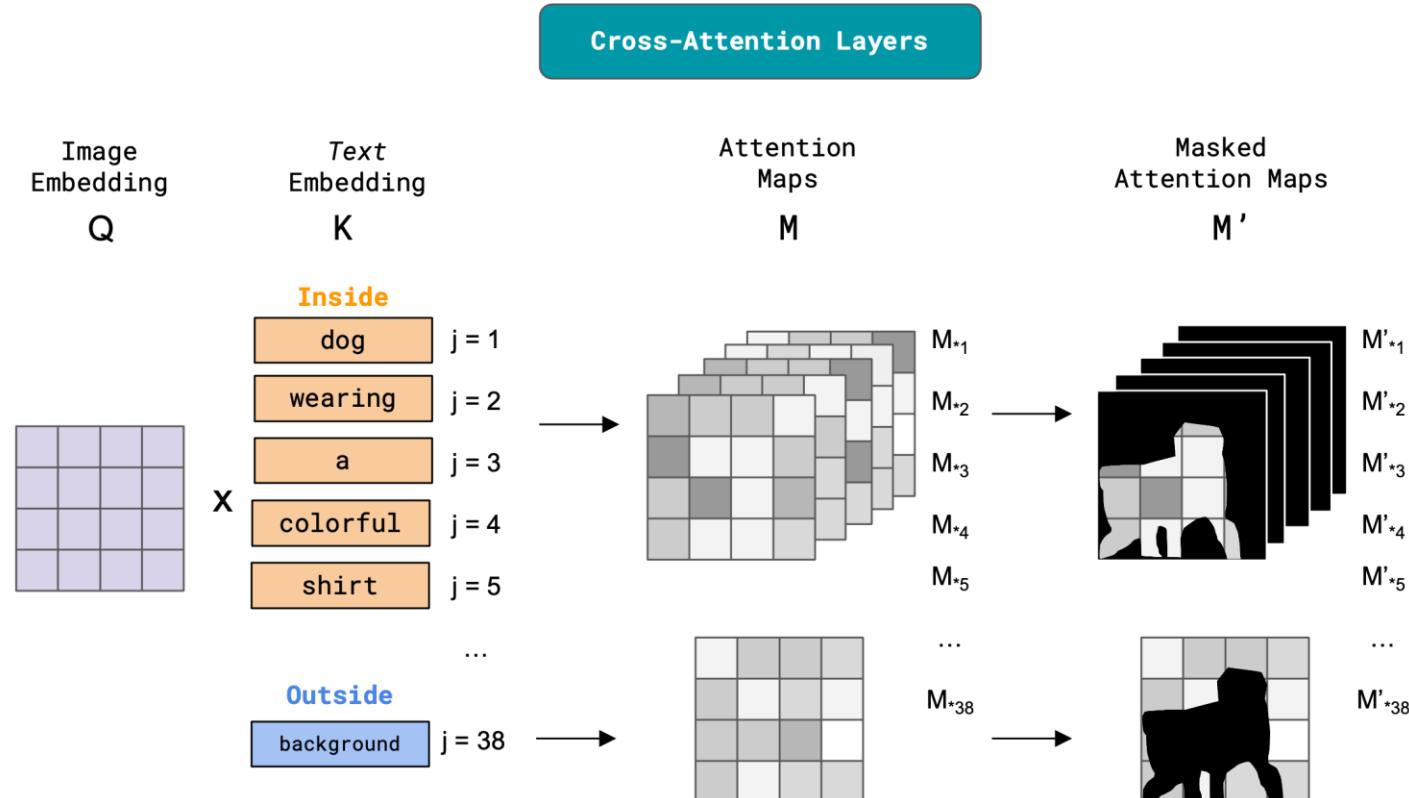
# Shape Guided Diffusion with Inside-Outside Attention\*



\*[2212.00210] Shape-Guided Diffusion with Inside-Outside Attention ([arxiv.org/](https://arxiv.org/))

Shape Guided Diffusion with Inside-Outside Attention ([shape-guided-diffusion.github.io](https://shape-guided-diffusion.github.io))

# Shape Guided Diffusion with Inside-Outside Attention\*

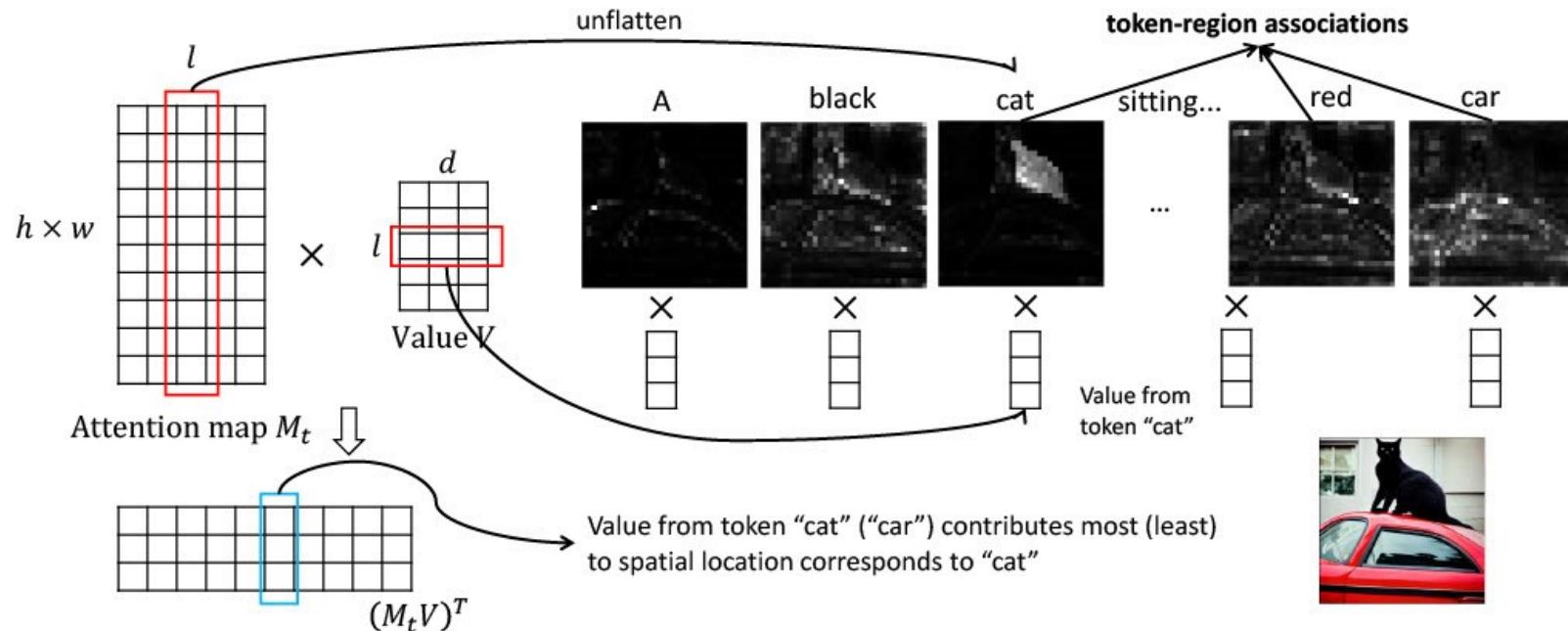


\*[2212.00210] Shape-Guided Diffusion with Inside-Outside Attention ([arxiv.org/](https://arxiv.org/))

Shape Guided Diffusion with Inside-Outside Attention ([shape-guided-diffusion.github.io](https://shape-guided-diffusion.github.io))

# Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis\*

## Cross-attention control



\*[2212.05032] Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis (arxiv.org)  
StructuredDiffusion (weixi-feng.github.io)

# Training-Free Structured Diffusion Guidance for Compositional Text-to-Image Synthesis\*

## Structured Diffusion Guidance

Prompt: A room with blue walls and a white sink.

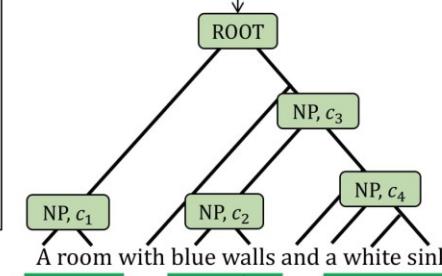
$\text{Parser } \xi$

Structured Representations

Constituency Tree

Scene Graph

$\text{CLIP}_{\text{text}}$



Condition Encoding Stage

$\mathcal{W}_p$   
A room with blue walls and ...

$\mathcal{W}_2$

<bos> blue walls <pad> ...

$f_k; f_v$

$K_p$

$f_v$

2D feature maps

Query  $Q^t$   
(unflatten for demo)

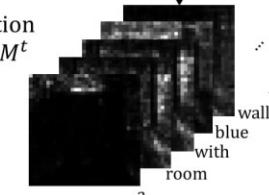
Attention  
maps  $M^t$

$V_p$   
 $V_1$   
 $V_2$   
 $V_3$   
 $V_4$

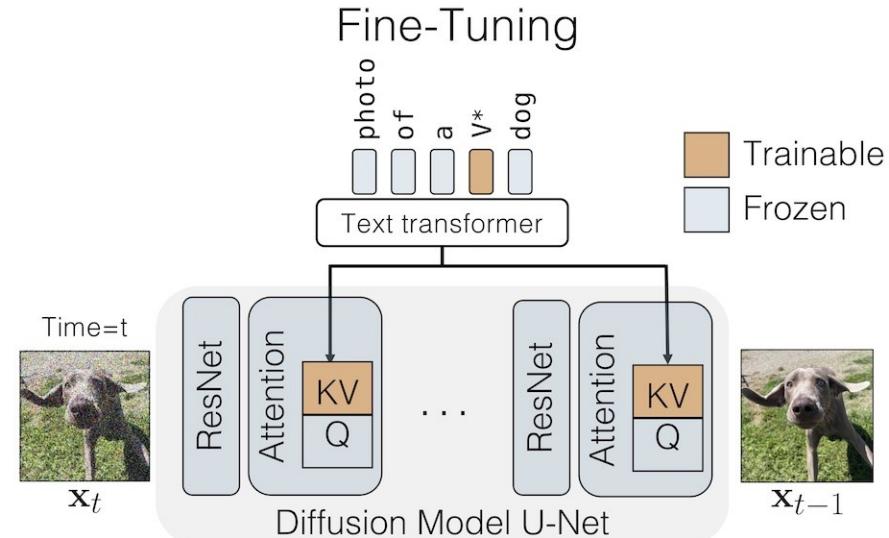
$$O^t = \frac{\sum_i M^t V_i}{5}$$

Sequence Alignment

Cross Attention Layers



# Multi-Concept Customization of Text-to-Image Diffusion\*



\*[2212.04488] Multi-Concept Customization of Text-to-Image Diffusion (arxiv.org)

Multi-Concept Customization of Text-to-Image Diffusion (cmu.edu)